DOCUMENT RESUME

ED 388 273 IR 017 422

AUTHOR Martins, Joaquim Arnaldo; Pinto, Joaquim Sousa
TITLE Hypermedia Courseware and Collaborative Tools for MS-Windows Environment.
PUB DATE 94
NOTE 7p.; In: Educational Multimedia and Hypermedia, 1994. Proceedings of ED-MEDIA 94--World Conference on Educational Multimedia and Hypermedia (Vancouver, British Columbia, Canada, June 25-30, 1994); see IR 017 359.
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Computer Software Development; *Cooperative Learning; *Courseware; Editing; Foreign Countries; Group Instruction; *Hypermedia
IDENTIFIERS Browsing; Microsoft Windows; *Object Oriented Programming

ABSTRACT

This paper describes the design of a set of hypermedia tools intended to produce courseware to be used for individual or group learning in a cooperative way. The system was developed for the MS-Windows environment, supporting OLE (object linking and editing) technology. With OLE technology, the user does not have to leave an application to create or edit specific data. The Hypermedia Editor was designed to create and annotate the courseware in a stand-alone mode; the Hypermedia Player allows students to browse documents for self or group study; and the Hypermedia Browser works jointly with the Editor and Player to view lessons or to build the hypermedia structure. Future developments in the area of document distribution and storage will require collaborative tools to make these systems more usable and ergonomic. (Contains nine references.) (AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

Hypermedia Courseware and Collaborative Tools for MS-Windows Environment

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

JOAQUIM ARNALDO MARTINS & JOAQUIM SOUSA PINTO
University of Aveiro / INESC
Campo Universitário, 3800 Aveiro, Portugal
Email: jam@inesca.pt

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Gary H. Marks
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)"

ED 388 273

Abstract: This paper describes the design of a set of hypermedia tools intended to produce courseware, to be used for self learning or distance learning and training environments in a cooperative way. These tools has roots in an standalone hypermedia editor developed to create courseware, to which where added other tools to enhance it´s usabilty, namely an hypermedia player to view courseware in self study mode or in groups, an hypermedia browser to create, to navigate or to display the hypermedia document structure and some tools to allow cooperative work. The underlying metaphors are the overhead projector, the lesson and the transparency. This system was developed for the MS-Windows environment, supporting OLE technology.

Nowadays the learning process is not confined to the childhood and school. Professional training, retraining and specialization involves a large number of people. The geographical spreading of these people raises the necessity to have distributed learning environments supported by intelligent tutoring systems were individuals or organizations can access to get self-learning information. Hypermedia systems (Nielsen, 1990) have enormous potential for training and education in scientific and non-scientific areas.

Rather than individual or self-learning it is important to create central learning and training centers that learners can access for learning and group study. Bearing this in mind, we are trying to setup some experiencies for cooperative group study and teaching, persuading the teachers to produce their teaching material in hypermedia support, instead of the traditional material (transparencies and lithographed material). Achieved this point, will be possible to the students to access the lesson's material, starting self-study or group study and, with the teachers support, to create lecture sessions. To support this environment we developed some hypermedia tools, able to handle several kinds of multimedia objects (text, sound, images and video) to produce, to organize and to play courseware in standalone or in group mode, as well as some underlying architecture to support these computer based cooperative activities.

Since we used the *OLE technology* to develop the described hypermedia tools we will introduce it´s basic concepts. The central philosophy behind OLE technology is to change, from a traditional application centered view of computing, towards a document centered view (Microsoft, 1992). This implies that a user does not have to leave an application to create or edit application specific data but instead, more comfortably, simply launch an application with a mouse selection on the specific data. In the OLE scheme, applications can either be clients or servers. *Server applications* are programs where linked or embedded data is created. *Client applications* are programs that receive OLE data that is linked or embedded. Each OLE object has a graphical representation and a behavior. The OLE client controls the OLE object graphical representation and is responsible for the shape assumed by the object in the document. The OLE Server handles the OLE object behavior through its *verbs*.

Underlying Collaborative Architecture

In centralized training and tele-teaching systems there is a conference server that deals with the social rules and with the database needs (Derycke, 1993). In such a system, the participants in a conference should know, in advance, what is its server location and what is the local library. People during a conference can not access to other libraries nor can meet at different places. For big conferences this scenario fits well, but for small

IR017432

BEST COPY AVAILABLE

conferences where it is needed some flexibility, it should be possible for the various participants to decide about the best room to meet and to access to all documents of each participant. Each place must be able to host at any time a conference and so it must support a series of tools to help in the organization, preparation and execution of these meetings. Such a distributed system as some advantages, namely: the independence of a specific machine or network connection (it is possible to choose another place to meet) and the possibility of having several conferences simultaneously over the same network, providing that groups are independent. The disadvantage is that the client must be prepared to play also the *server role* (when selected as a host) and must deal with the problem of replicated objects which belong to a group of people in an efficient way (Abdel, 1988), since each user must have a copy of the working document.

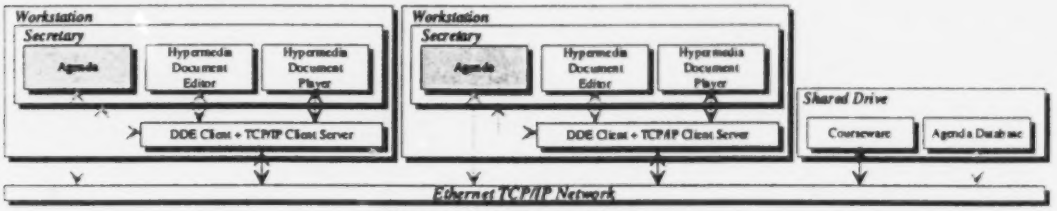

Figure 1. Proposed architecture

To implement the scenario shown in Figure 1, we used the MS-Windows environment taking profit of the OLE technology. For now, we built the following group of tools: *Hypermedia Editor*, *Hypermedia Player* and *Hypermedia Browser*. The Hypermedia Editor was designed to produce and annotate the courseware, for now in stand-alone mode. The Hypermedia Player was designed for self-study, learning and group study. The Hypermedia Browser works jointly with the Editor and Player and may be used to browse the lessons or to build the hypermedia structure. *Secretary* and *Agenda* tools will be the engine for the proposed architecture. *Shared Drive* is implemented through a MS-DOS file system extension (Novell Netware, NFS, Windows for Workgroups, etc).

The main difficulty to design a Cooperative Hypermedia Editor, Browser or Player for this environment concerns with the granularity imposed by OLE technology. It is confined to the object level since we can not have the control of OLE servers. This is not a severe problem since are the authors that control the level of object granularity and from the pedagogical point of view, the transparencies should present clear ideas and key points without being overcrowded with information, which may be confusing. They should use simple objects without to much complexity, which is consistent with the metaphor of a transparency used to design the system.

Hypermedia Editor, Browser and Player Metaphors

A full technical description of the *Hypermedia Editor* describing the characteristics and functionality's can be seen in Pinto (1994). For an ergonomic analysis and a detailed description of the advantages and disadvantages of the use of OLE technology in hypermedia see Pauwels (1993, 1994). Just to understand the metaphors used in the design of the hypermedia engine they will be described, briefly, in the following paragraphs.

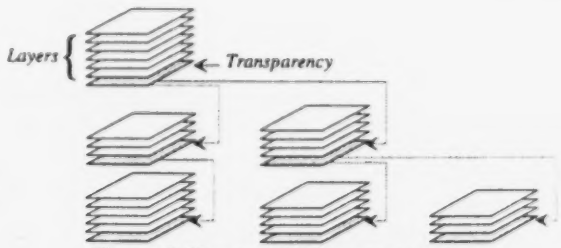

Figure 2. Hypermedia document structure

The metaphors used to describe the applications are the *hypermedia overhead projector* on which *lessons*, composed of piles of *hypermedia transparencies*, can be displayed either in a simple sequential order or in a

BEST COPY AVAILABLE

more sophisticated one, so called "web" structure, as shown in Figure 2. One hypermedia node, consists of a *transparency* which may, or may not, be covered by one or more *overlaying layers*. A transparency is composed by *multimedia objects* and linked through *anchors* with other transparencies or layers. A transparency yields anchors and multimedia objects; the layers only hold multimedia objects.

The hypermedia engine admits links between transparencies, layers and other lessons. There is no need of a script language to create or to control the engine since the transparency and layer insertion and the anchoring among them is constructed automatically through the user actions.

An anchor establishes a link among a source and a destination point. The result of selecting an anchor is to follow up the link to its destination, that may be another transparency, a *superimposed layer*, an *alternated layer* or even another lesson. A multimedia object is any embedded or linked OLE object and may be time-variant or change its value. It is therefore a *hyperlink* to the associated server, which implies that an object activation may result in a video sequence presentation.

Hypermedia Browser

In very complex hypermedia lessons it is easy for a learner to get "lost" inside the lesson. Therefore it is necessary to have a structure visualization to help the user to navigate through, or browse, the lesson (Dumais, 1988; Nielsen, 1990). Such tool may also be used as a cooperative authoring tool used by several teachers to define the structure of the lessons; afterwards each teacher can edit his set of transparencies. The granularity level is confined to the object level, in editing mode, since to edit an object the control passes to the OLE server application which puts some problems to work in a cooperative way. The authors can cooperate on the lesson structure definition and object placement but they will be able only to edit single objects in an individual way, which we do not see as a restriction. The *Hypermedia Browser* presents the lesson structure independently of its content. Figure 3 shows the interaction between the Browser, the Editor and the Player, in stand-alone mode. In the Browser window, each transparency is identified by an icon and its name, and the connections among transparencies are represented by arrows.

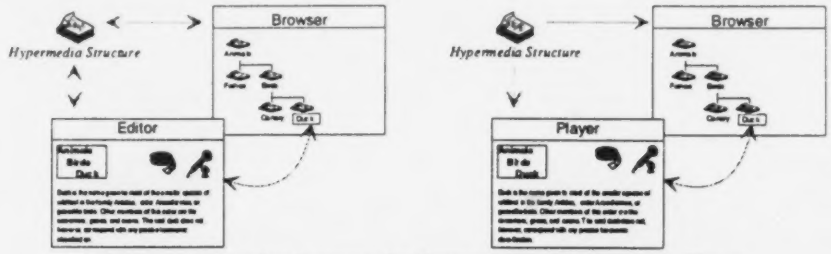

Figure 3. Interaction between the Browser, the Editor and the Player

To present the lesson structure, the browser linearizes and places it in a *virtual grid* using an algorithm that recalculates the lesson network each time its structure changes. The links in the network are calculated using the anchors among transparencies as reference. Some paths may be difficult to identify in intricate structures. To improve this and to allow a better understanding of the network, the browser draws the links in eight colors, that users may select or deselect. Besides the color mechanism, a user can always know the entire list of connections and the associated layers of a transparency simply through a mouse selection on it.

Using the Browser to build lessons

Any user, without programming skills can construct complex lessons, because the process is entirely done visually. A user inserts new blank transparencies and then anchors them with each other through a dialogue box that shows the possible link destinations. Each time a user inserts a new anchor among transparencies, the Browser recalculates the network. Depending on the depth of the source transparency in the network, it places a visible anchor on left or on the right side of the source transparency. Usually the network has bi-directional links. To simplify the presentation and make it more readable, bi-directional links are drawn through a bi-directional line.

BEST COPY AVAILABLE

Using the Browser to browse information

The other Browser functionality is to allow the user to browse through the lesson web structure. The currently displayed transparency on the overhead projector is identified in the Browser window by a shaded look. To jump into another transparency in the Editor or Player, a user should simply, in the Browser window, click over the desired transparency. The Browser sends an order to the Editor or Player to display the selected transparency. This behavior is very useful if the user is "lost" in the structure and wants to go to a "known" point.

Hypermedia Player

To play the lessons it is not needed a full Editor. The *Hypermedia Player* tool allows the students to browse documents in self learning mode or, in a cooperative environment, controlled by a teacher or a group moderator.

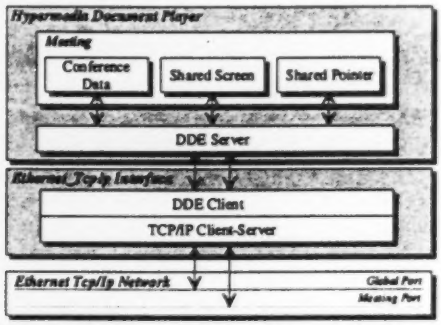

Figure 4. Hypermedia Player Internal Architecture

In the Player, the students are not able to modify or annotate the lessons and have a simple mechanism to go through the transparencies.

The Cooperative Player

The architecture for cooperation between the students and teachers is built on top of a TCP/IP network. The communication structure is not based on the client/server classical model. Instead, any participant has the possibility to host a meeting, i.e., each station is simultaneously client and server. This means that does not exist any global server able to deal with the global data, such as the name of all running meetings. Each meeting has a *proponent* that is responsible for the *meeting data* and to play the server role. The meeting data has three components: *Conference Data*, *Shared Screen* and *Shared Pointer*. The Conference Data includes the name of all running conferences and when the user is attached to a specific one, the name of all elements in the conference. The Shared Screen holds the size and scroll bar positions of the public overhead projector. The Shared Pointer section has the queue of users for the shared pointer, the shared pointer position and the identification of the actual owner, if any.

Due to the non existence of a central server to provide information about meetings, we created a *global channel* listened by all users. Always a user plans to connect to an existent meeting, sends a network message, through the global channel, requesting information about running meetings. Each proponent replies with the meeting name and with the *meeting channel*, that is the channel through which the conference specific data flows.

After a user connects to a meeting, the proponent synchronizes the new station. This involves the new user local screen sizing, because all partners should see the same thing at the same time (WYSIWIS), and the delivering of all conference data. Each user can have a request for the global pointer possession in the global pointer queue, that follows the "first come first goes" algorithm. The proponent has the privilege to change the user's order in that queue and revoke the pointer possession from a user. The actions performed by the global pointer owner, such as a OLE object activation or a window scrooling, are reflected in all stations in a meeting.

A user, unless is the global pointer owner, can leave the meeting at any time. The meeting finishes when the proponent leaves the meeting.

Figure 5 shows the look and feel of the hypermedia editor, browser and player.

BEST COPY AVAILABLE

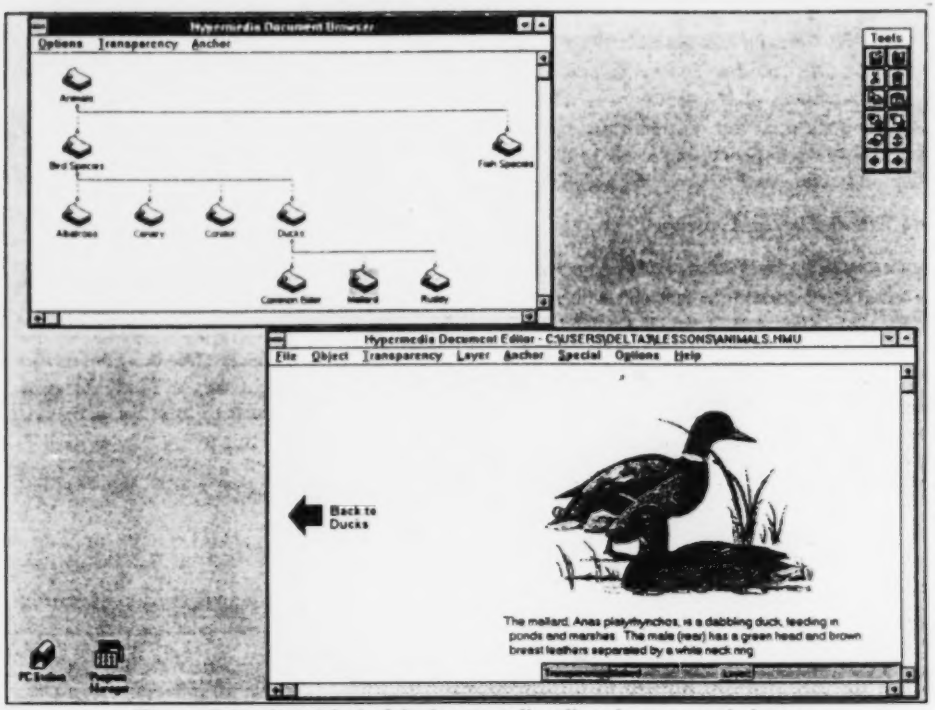
Options Transparency Anchor
Tools
Hypermedia Document Editor - C:\USERS\DELTA\LESSONS\ANIMALS.HMU
File Object Transparency Layer Anchor Special Options Help
Back to Ducks
The mallard, Anas platyrhynchos, is a dabbling duck, feeding in ponds and marshes. The male (rear) has a green head and brown breast feathers separated by a white neck ring.

Figure 5. Look and feel of the hypermedia editor, browser and player

Future Work on the Distributed Cooperative Environment

Now we discuss some ideas to enhance the work in a cooperative way. Some of them are already on going work, while others are ideas we starting to explore, namely the transmission of sound and image over TCP/IP protocols in a synchronous and asynchronous way, which will improve the quality of computer supported collaborative work over TCP/IP networks.

Associated with this cooperative environment it is needed to have special attention to aspects like document distribution and storage and to have some tools to make these systems more usable and ergonomic.

Collaborative Tools

Some of the tools needed in our environment are already used in other collaborative environments (Derycke, 1993), namely a *meeting room*, a *vote* mechanism to help to take decisions, a *timer* to control the duration of talks, pointer possessions and meetings. Mary, a blond blue eyes *secretary*, an *agenda*, a *briefcase* to carry personal documents to the meetings, a *synchronous sound server* to distribute sound/voice to all participants and an *asynchronous sound and video server* are other tools included into our environment.

Under current implementation there are the sound and image servers, the agenda and the voting mechanism. Sound and video are carried over IP networks and the servers are software based and will control the conversation among a small group of people. Some tests done in our TCP/IP/Ethernet network, and using a Sound Blaster board as the input and output device for voice, shown that until a network load of 60-70%, depending on several configuration factors, the users don't detect any degradation in the voice signal for packet losses of 2% (Azevedo, 1994). Now we are studying some form of asynchronous communication with voice and even appended video, based on the fact that in some scenarios the conversation is done in message bursts.

In the *Agenda* the users can see a list of booked meetings, create new meetings, adhere to a meeting or to see meeting details, such as the date and time, participant names, requirements, etc. It is possible to create public meetings, in which all users can participate, and private meetings. In this case, only the participants included in the participants list are allowed to participate or see the meeting details. The meeting requirements contain a list of the meeting files, that may be placed by Mary in each user briefcase before the meeting start.

BEST COPY AVAILABLE

Besides the Agenda, it is possible in any moment during a meeting to admit new users to the group. In this case, the meeting proponent will be responsible to admit or not admit the new user to the meeting or to submit the problem to all members of the group by a voting process. This process is done using the *Vote* tool. The vote process may be open or secret. In first case, each participant knows the vote of each other and the result is done through the difference between the votes. In the last case the participants only know the result. This an example of the utility of the vote tool that may be used in many other situations during the meeting.

Document distribution and storage

Hypermedia documents may have an enormous amount of data, which puts several problems such as large databases and long transmission times. It will be useful to use compression and pre-fetching techniques to reduce storage space and distribution time. The pre-fetching can be used to distribute documents before the meeting. The *Secretary* looks up to the *Agenda* and sometime before the meeting start, fits the boss *briefcase* with the conference stuff. During the meeting the pre-fetching technique may be used to load the next possible transparencies to minimize the time between transparencies. It is also important to look for distributed storage systems, since there are documents spread over several rooms and libraries. But to have organized libraries of multimedia courseware and to manipulate and access them in an efficient way it is needed to look for distributed multimedia databases and high level APIs able to handle different types of databases and/or file systems.

Conclusions

We described a set of hypermedia tools implemented under the MS-Windows environment to browse and play hypermedia documents in standalone or in a cooperative way, using the OLE technology, which open the possibility to integrate into the hypermedia documents, objects produced by any MS-Windows OLE server application. This allows the users to use their familiar applications to produce, annotate or play the courseware, which, jointly with the adoption of overhead projector and transparency metaphor's, results in a simple and ergonomic user interface.

References

Abdel-Wahab, H.M. & Guan, S. & Nievergelt, J., (1988). "Shared Workspaces for Group Collaboration: An Experiment Using Internet and UNIX Interprocess Communications", in *IEEE Communications Magazine*, November, 10-16

Azevedo, J. & Martins, J.A.& Oliveira, P.G. (1994). "An Experimenta Study on Voice/Data Integration over an Ethernet LAN", in *Proccedings of INDC'94 - International Conference on Information Networks and Data Communications*, April 18-21, Madeira, Portugal.

Derycke, A.C. & Kaye, A.R (1993). "Participative modelling and design of collaborative distance learning tools in CO-LEARN project", in *Proceedings of the IFIP TC3 Third Teleteaching Conference*, 191-200.

Dumais, S., (1988). "Handbook of Human Computer Interaction", in *Textual Information Retrieval*, Ed. Martin Helander, North-Holland.

Microsoft Press (1992), "Object Linking and Embedding - Programmer's Reference".

Nielsen, J.(1990). "Hypertext and Hypermedia", *Academic Press.*

Pauwels, H.W.J.B. & Pinto, J. S. & Santos, B. S. & Martins, J. A. (1993). "An Integrated Courseware Editor Based on OLE Technology", in *Lectures Notes In Computer Science*, 733, 439-440.

Pauwels, H.W.J.B. & Pinto, J. S. & Santos, B. S. & Martins, J. A. (1994). "Integrating Existing Applications in Hypermedia Learning Material (General Issues & Experiences with OLE technology)", in *Proccedings of ED-MEDIA 94 - World Conference on Educational Multimedia and Hypermedia*, June 25-29, Vancouver, Canada.

Pinto, J. S. & Pauwels, H.W.J.B. & Martins, J. A. & Santos, B. S. (1994). "HyDE: a hypermedia document editor based on OLE technology", in *Proccedings of ICMCS'94 - International Conference on Multimedia Computing*, May-15-19, Boston, MA, U.S.A..

Acknowledgements

We would like to thank Hubert B. Pauwels and Beatriz S. Santos who actively collaborated in the design of the standalone document editor. This editor was also funded by the european DELTA project CO-LEARN (D2005).